

## IMAGE TYPE CLASSIFICATION USING EDGE FEATURES

### Background of the Invention

[0001] The invention relates to image processing. It finds particular application in conjunction with classification of images between natural pictures and synthetic graphics, and will be described with particular reference thereto. However, it is to be appreciated that the invention is also amenable to other like applications.

[0002] During the past several decades, products and services such as TVs, video monitors, photography, motion pictures, copying devices, magazines, brochures, newspapers, etc. have steadily evolved from monochrome to color. With the increasing use of color products and services, there is a growing demand for "brighter" and more "colorful" colors in several applications. Due to this growing demand, display and printing of color imagery that is visually pleasing has become a very important topic. In a typical color copier application, the goal is to render the scanned document in such a way that it is most pleasing to the user.

[0003] Natural pictures differ from synthetic graphics in many aspects, both in terms of visual perception and image statistics. Synthetic graphics are featured with smooth regions separated by sharp edges. On the contrary, natural pictures are often noisier and the region boundaries are less prominent. In processing scanned images, it is sometime beneficial to distinguish images from different origins (e.g., synthetic graphics or natural pictures), however, the origin or "type" information about a scanned image is usually unavailable. The "type" information is extracted from the scanned image. This "type" information is then used in further processing of the images. High-level image classification can be achieved by analysis of low-level image attributes geared for the particular classes. Coloring schemes (e.g., gamut-mapping or filtering algorithms) are tailored for specific types of images to obtain quality reproduction. Once an image has been identified as a synthetic graphic image, further identification of image characteristics can be used to fine-tune the coloring schemes for more appealing reproductions. The most prominent characteristics of a synthetic graphic image include patches or areas of

the image with uniform color and areas with uniformly changing colors. These areas of uniformly changing color are called sweeps.

[0004] Picture/graphic classifiers have been developed to differentiate between a natural picture image and a synthetic graphic image by analyzing low-level image statistics. For example, U.S. Patent No. 5,767,978 to Revankar et al. discloses an adaptable image segmentation system for differentially rendering black and white and/or color images using a plurality of imaging techniques. An image is segmented according to classes of regions that may be rendered according to the same imaging techniques. Image regions may be rendered according to a three-class system (such as traditional text, graphic, and picture systems), or according to more than three image classes. In addition, two image classes may be required to render high quality draft or final output images. The image characteristics that may be rendered differently from class to class may include half toning, colorization and other image attributes.

[0005] Synthetic graphics are typically generated using a limited number of colors, usually containing a few areas of uniform colors. On the other hand, natural pictures are more noisy, containing smoothly varying colors. A picture/graphic classifier can analyze the colors to distinguish between natural picture images and synthetic graphic images.

[0006] Synthetic graphic images contain several areas of uniform color, lines drawings, text, and have very sharp, prominent, long edges. On the other hand, natural pictures are very noisy and contain short broken edges. A picture/graphic classifier can analyze statistics based on edges to distinguish between natural picture images and synthetic graphic images.

[0007] Classifiers that can be used to solve a certain classification problem include statistical, structural, neural networks, fuzzy logic, and machine learning classifiers. Several of these classifiers are available in public domain and commercial packages. However, no single classifier seems to be highly successful in dealing with complex real world problems. Each classifier has its own weaknesses and strengths.

[0008] The invention contemplates new and improved methods for classifying images that overcome the above-referenced problems and others

### Summary of the Invention

[0009] A method and system for classifying images between natural pictures and synthetic graphics is provided. In embodiments of the invention, a picture/graphic classification method and system implements edge features for image classification of natural pictures and synthetic graphics. In another embodiment of the invention, a picture/graphic combination classification method and system using edge features, color discreteness features, and SGLD texture features is used to classify images between natural picture and synthetic graphic classes.

### Brief Description of the Drawings

[0010] The invention may take form in various components and arrangements of components, and in various steps and arrangements of steps. The drawings are for purposes of illustrating embodiments of the invention and are not to be construed as limiting the invention.

[0011] FIG. 1 is a flowchart of an image classification process using SGLD texture features in an embodiment of the invention;

[0012] FIG. 2 is a more detailed flowchart of the SGLD matrix initialization and construction process portion of the flowchart in FIG. 1,

[0013] FIG. 3 is a flowchart of an image classification process using one-dimensional (1-D) color discreteness features in an embodiment of the invention;

[0014] FIG. 4 is a flowchart of an image classification process using two-dimensional (2-D) color discreteness features in an embodiment of the invention;

[0015] FIG. 5 is a flowchart of an image classification process using three-dimensional (3-D) color discreteness features in an embodiment of the invention,

[0016] FIG. 6 is a flowchart of an image classification process using a combination of 1-D color discreteness features, 2-D color discreteness features, and 3-D color discreteness features in an embodiment of the invention;

[0017] FIG. 7 is a flowchart of an image classification process using edge features in an embodiment of the invention;

[0018] FIG. 8 is a flowchart of an image classification process using a combination of SGLD texture features, color discreteness features, and edge features in an embodiment of the invention; and,

[0019] FIG. 9 is a block diagram of an image processing system using a “binary” image classification process (i.e., classification of images between natural picture or synthetic graphic classes).

#### **Detailed Description of the Invention**

[0020] Spatial gray-level dependence (SGLD) techniques for image analysis are well known. SGLD feature extraction creates a 2-D histogram that measures first and second-order statistics of an image. These features are captured in SGLD matrices. This was originally proposed for texture analysis of multi-level images. Additionally, since texture features distinguish natural pictures from synthetic graphics, SGLD techniques can be applied to picture/graphic classification of images. A picture/graphic classifier can be created with algorithms that analyze the texture features captured in SGLD matrices. Using the SGLD texture features, the classifier works to determine whether a scanned image is a natural picture or synthetic graphic. Furthermore, in color images, the luminance component typically contains enough information to determine the origin of the image. Therefore, an SGLD matrix that captures the luminance component of an image and a picture/graphic classifier using the luminance component from the matrix in a classification algorithm can determine whether the image is a natural picture or synthetic graphic.

[0021] With reference to FIG. 1, a flowchart of an image classification process 100 using SGLD texture features in an embodiment of the invention is shown. Generally, the classification process filters an input image to smooth out halftones, builds an SGLD matrix from the smoothed image, extracts texture features from the matrix, and performs an algorithm to determine whether the image is a natural picture or synthetic graphic based on one or more of the texture features.

[0022] More specifically, the process 100 begins with an input image 102. The image is processed using a low-pass filter 104 (e.g., a  $W \times W$  averaging filter) to smooth the luminance component and reduce any halftone noise. The SGLD matrix is

basically a  $GL \times GL$  2-D histogram, where  $GL$  is the number of gray levels (e.g., 256). The SGLD matrix is generated by first performing an initialization (e.g., set to zero) 106. Next, the SGLD matrix is built from the smoothed image 108. The SGLD matrix is a 2-D histogram corresponding to certain characteristics of the pixels in the input image. For each pixel  $(m, n)$  in the smoothed image, a neighboring value is calculated using the following logic and equations:

if  $|x(m, n + d) - x(m, n)| > |x(m + d, n) - x(m, n)|$

then  $y(m, n) = x(m, n + d)$ ,

otherwise  $y(m, n) = x(m + d, n)$ , (1),

where  $x(m, n)$  is the smoothed pixel value at  $(m, n)$ ,  $(m, n + d)$  and  $(m + d, n)$  are vertical and horizontal neighbors, respectively, and  $d$  is a fixed integer (typically one or two)

[0023] With reference to FIG. 2, a flowchart of an embodiment of the SGLD matrix initialization and construction process is shown. The initialization step 106 sets the SGLD matrix to zero and sets a pixel counter ( $N$ ) to zero 154. The SGLD matrix is constructed from a low-pass filtered image 152 provided by the low-pass filter 104. Construction of the SGLD matrix begins by getting a pixel  $(m, n)$  156 from the filtered image. A neighboring value for the pixel  $(m, n)$  is calculated using the algorithm in equation (1). If  $|x(m, n + d) - x(m, n)| > |x(m + d, n) - x(m, n)|$  158, then  $y(m, n) = x(m, n + d)$  160. Otherwise,  $y(m, n) = x(m + d, n)$  162. As is apparent, if pixel  $(m, n)$  is in a flat area where  $x(m, n)$  is equal to  $y(m, n)$ , the entry  $[x(m, n), y(m, n)]$  is on the diagonal. On the other hand, if  $(m, n)$  is on an edge, the difference between  $x(m, n)$  and  $y(m, n)$  will be significant, and  $[x(m, n), y(m, n)]$  will be far away from the diagonal.

[0024] The entry  $[x(m, n), y(m, n)]$  in the SGLD matrix is then increased by one and the pixel counter ( $N$ ) is increased by one. Next, a check is made to determine if the calculation was for the last pixel 166 of the input image. If so, SGLD matrix construction is complete and the SGLD matrix is ready for feature extraction 168. Otherwise, the next pixel is retrieved 156 from the input image.

[0025] For the matrix, the neighboring pixels in synthetic graphic images are expected to be either correlated or very different. In other words, for synthetic graphic images, SGLD matrix entries are usually either on the diagonal or far away from the diagonal. This is because most pixels are either at the flat regions or on the edges. On

the other hand, pixels of natural pictures are not expected to have many abrupt changes. Accordingly, masses are expected to be concentrated at the entries that are near the diagonal for natural picture images. This shows the noisy nature of the natural picture images.

[0026] Returning to FIG 1, many features (e.g., variance, bias, skewness, fitness) can be extracted from the SGLD matrix to classify the input image between natural picture and synthetic graphic classes. The features can be implemented individually or combined in various methods (e.g., linear combination). Once the SGLD matrix is built, a feature or combination of features is selected for extraction 110 and processed using feature algorithms. For example, a first feature algorithm measures variance (V) (i.e., the second-order moment around the diagonal) 112 and is defined as

$$V = \sum_{|n-m| > \Delta} s(m, n) (m - n)^2 / N \quad (2),$$

where  $s(m, n)$  is the  $(m, n)$ -th entry of the SGLD matrix,  $\Delta$  is an integer parameter typically between 1 and 16 and;

$$N = \sum_{|n-m| > \Delta} s(m, n) \quad (3)$$

[0027] As the summation is over all  $(m, n)$  such that  $|m - n| > \Delta$ , all the pixels in the flat regions are ignored. For synthetic graphic images, the remaining pixels are on the edges, while for natural picture images, both pixels in the noisy regions and pixels on the edges are included. Variance (V) is typically larger for synthetic graphic images than for natural picture images

[0028] The second feature algorithm measures average bias (B) 114 and is defined as:

$$B = \sum_{|n-m| > \Delta} s(m, n) [n - \mu(m)]^2 / N \quad (4),$$

where  $\mu(m)$  is the mean of  $s(m, n)$  for a fixed  $m$ . For a given  $m$ , the distribution of  $s(m, n)$  is roughly symmetrical about the diagonal for natural picture images, as noise typically has a zero mean symmetrical distribution. As a result B is usually small for natural picture images. For synthetic graphic images,  $s(m, n)$  is usually unsymmetrical and B is large.

[0029] The third feature algorithm measures skewness (S) 116 and is defined as:

$$S = \text{skewness} = \frac{\sum_{n=0}^{GL-1} \left| n - m \right| (n - m) s(m, n)}{\sum_{m=0}^{GL-1} \left| n - m \right| s(m, n)} \frac{1}{c(n)/C}, \quad (5),$$

where:

$$c(n) = \sum_{m=0}^{GL-1} s(m, n) \text{ and } C = \sum_{n=0}^{GL-1} c(n). \quad (6)$$

[0030] The fourth feature algorithm measures fitness (F) 118 and is defined to be:

$$F = \text{fitness} = \frac{\sum_{n=0}^{GL-1} (n - m)^2 s(m, n)}{\sigma^2}, \quad (7),$$

where  $\sigma$  is defined such that

$$\sum_{d=0}^{\sigma} [s(m, m + d) + s(m, m - d)] = 0.6x \quad (8)$$

[0031] The image type decision 120 compares the result of the feature algorithm(s) to previously selected low and high thresholds (i.e.,  $T_L$  and  $T_H$ , respectively) depending on the algorithm(s) and combinations selected. If the result of the feature algorithm(s) is below the low threshold ( $T_L$ ), the image is classified as a natural picture 122. If the result exceeds the high threshold ( $T_H$ ), the classification is synthetic graphic 126. Obviously, if the behavior of a particular feature is converse to this logic, the decision logic can be easily reversed to accommodate. If the result of the feature algorithm(s) is equal to or between the low and high thresholds, the class of the image cannot be determined (i.e., indeterminate 124) from the feature or combination of features selected. It is understood that a number of other alternatives are possible. For example, a result equal to a particular threshold can be said to be determinative of the image class, rather than indeterminate. Also, in certain circumstances the low and high threshold can be equal.

[0032] With reference to FIG. 3, a flowchart of an image classification process 200 using 1-D color discreteness features in an embodiment of the invention is shown. The process 200 begins with an input image 202. First, the input image is

transformed into a color space 204, in which the classification is performed. Although CIELUV space is used in the embodiment being described, many other color spaces can also be used. Next, the image is smoothed using an averaging filter 206 to remove any noise due to halftones. For example, a 4 x 4 filter was used successfully. Next, a 1-D color histogram is computed for a color channel (i.e., luminance (L), U, and V) 208, any combination of two 1-D color histograms may be computed, or all three 1-D color histograms may be computed, depending on the algorithm or combination of algorithms to be performed for classification.

[0033] Next, the histogram(s) may be pre-filtered using a low-pass filter, for example an averaging filter 209 to smooth out artificially boosted probabilities due to the size of the sample. The averaging filter, for example, works particularly well for images with a relatively small sample size that lead to a sparse distribution over the histogram. Next, the histogram(s) may be pre-adjusted by applying an F(H) function to histogram entries with high bin counts representing large areas 210 in the image in the same color. Predetermined thresholds define high bin counts and determine whether or not the F(H) function is applied to the histogram bin. The F(H) function is a monotonically decreasing function (e.g., cube root) and under-weights the histogram counts for the large areas. This under-weighting, for example, works particularly well to correct a bias to incorrectly classify pictures with large flat regions (e.g., sky, paper background) as synthetic graphic images.

[0034] The L, U, and/or V histograms are normalized 211 by the number of pixels in the image. The color representation scheme is invariant under rotation and translation of the input image and the normalization provides scale invariance. If  $I(i)$  is the histogram of an image, where the index  $i$  represents a histogram bin, then the normalized histogram  $H$  is defined as follows:

$$H(i) = \frac{I(i)}{\sum_{i=0}^{GL-1} I(i)} \quad (9)$$

[0035] Since synthetic graphics are generated using a limited number of colors, synthetic graphic images usually are comprised of a few areas of uniform color. Hence, the color histograms for a synthetic graphic image usually contain several sharp

peaks. On the other hand, natural pictures usually contain more colors with smoothly varying transitions. Hence, natural pictures are more noisy and produce histograms containing fewer and smoother peaks. This difference in the histograms is captured in color discreteness algorithms for each color channel (i.e., R\_L algorithm 212, R\_U algorithm 214, and R\_V algorithm 216). The color discreteness algorithms are defined as follows:

$$R_L = \sum_{x=1}^{GL-1} |H_L(x) - H_L(x-1)| \quad (10),$$

$$R_U = \sum_{x=1}^{GL-1} |H_U(x) - H_U(x-1)| \quad (11),$$

$$R_V = \sum_{x=1}^{GL-1} |H_V(x) - H_V(x-1)| \quad (12),$$

where GL is the number of bins in the H\_L, H\_U, and H\_V color histograms. The algorithms in equations (10) – (12) are derived from the 1-D color discreteness algorithm for any color space, defined for color channel s of a generic color space as follows:

$$R_s = \sum |H_s(x) - H_s(x-1)| \quad (13).$$

[0036] The image type decision 218 compares the results of the 1-D color discreteness algorithms (i.e., 212, 214, or 216) selected for performance to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of the selected 1-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124) by the 1-D color discreteness feature (i.e., R\_L, R\_U, or R\_V). Alternatively, the classifier may use all three 1-D color discreteness features in any sequence, any combination of two features in any sequence, or any one feature (as described above).

[0037] With reference to FIG 4, a flowchart of an image classification process 300 using 2-D color discreteness features in an embodiment of the invention is shown. Like the process 200 using 1-D color discreteness features, the process 300 begins with an input image 202, the input image is transformed into a color space 204, and

the image is smoothed using an averaging filter 206. Again, although CIELUV space is used in the embodiment being described, many other color spaces can also be used. Next, a 2-D color histogram is computed for two color channels (i.e., LU, LV, or UV) 308, any combination of two 2-D color histograms may be computed, or all three 2-D color histograms may be computed, depending on the algorithm or combination of algorithms to be performed for classification.

[0038] Next, like the process 200 using 1-D color discreteness features, the 2-D histogram(s) may be pre-filtered using an averaging filter 209 and may be pre-adjusted by applying an  $F(H)$  function to histogram entries representing large areas 210. Pre-filtering is important for the 2-D histogram(s), as they are typically sparse. Next, the LU, LV, and/or UV histograms are normalized 311 by the number of pixels in the image using equation (9).

[0039] Next, like the process 200 using 1-D color discreteness features, the difference between histograms of natural picture images and synthetic graphic images is captured in 2-D color discreteness algorithms (i.e.,  $R_{LU}$  algorithm 312,  $R_{LV}$  algorithm 314, and  $R_{UV}$  algorithm 316). The 2-D color discreteness algorithms for the LUV color space are defined as follows.

$$R_{LU} = \sum_{x=1}^{GL-1} \sum_{y=1}^{GL-1} \left( \left| H_{LU}(x, y) - H_{LU}(x-1, y) \right| + \left| H_{LU}(x, y) - H_{LU}(x, y-1) \right| \right) \quad (14),$$

$$R_{LV} = \sum_{x=1}^{GL-1} \sum_{y=1}^{GL-1} \left( \left| H_{LV}(x, y) - H_{LV}(x-1, y) \right| + \left| H_{LV}(x, y) - H_{LV}(x, y-1) \right| \right) \quad (15),$$

$$R_{UV} = \sum_{x=1}^{GL-1} \sum_{y=1}^{GL-1} \left( \left| H_{UV}(x, y) - H_{UV}(x-1, y) \right| + \left| H_{UV}(x, y) - H_{UV}(x, y-1) \right| \right) \quad (16),$$

where  $GL$  is the number of bins for each component in the  $H_{LU}$ ,  $H_{LV}$ , and  $H_{UV}$  color histograms. The algorithms in equations (14) – (16) are derived from the 2-D color discreteness algorithm for any color space, defined for color channel  $s$  and color channel  $t$  of a generic color space as follows

$$R_{st} = \sum \left( \left| H_{st}(x, y) - H_{st}(x-1, y) \right| + \left| H_{st}(x, y) - H_{st}(x, y-1) \right| \right) \quad (17)$$

[0040] The image type decision 318 compares the results of the 2-D color discreteness algorithm (i.e., 312, 314, or 316) selected for performance to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of a 2-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124) by the 2-D color discreteness feature (i.e., R\_LU, R\_LV, or R\_UV). Alternatively, the classifier may use all three 2-D color discreteness features in any sequence, any combination of two features in any sequence, or any one feature (as described above).

[0041] With reference to FIG. 5, a flowchart of an image classification process 400 using 3-D color discreteness features in an embodiment of the invention is shown. Like the process 200 using 1-D color discreteness features and the process 300 using 2-D color discreteness features, the process 400 begins with an input image 202, the input image is transformed into a color space 204, and the image is smoothed using an averaging filter 206. Again, although CIELUV space is used in the embodiment being described, many other color spaces can also be used. Next, a 3-D color histogram is computed for the three color channels (i.e., LUV) 408.

[0042] Next, like the process 200 using 1-D color discreteness features and the process 300 using 2-D color discreteness features, the 3-D histogram may be pre-filtered using an averaging filter 209 and may be pre-adjusted by applying an  $F(H)$  function to histogram entries representing large areas 210. Pre-filtering is important for the 3-D histogram, as it is typically sparse. Next, the LUV histogram is normalized 411 by the number of pixels in the image using equation (9).

[0043] Next, like the process 200 using 1-D color discreteness features and the process 300 using 2-D color discreteness features, the difference between histograms of natural picture images and synthetic graphic images is captured in a 3-D color discreteness algorithm (i.e., R\_LUV algorithm 414). The 3-D color discreteness algorithm is defined as follows:

$$R\_LUV = \sum_{x=1}^{GL-1} \sum_{y=1}^{GL-1} \sum_{z=1}^{GL-1} \left( \begin{array}{l} |H\_LUV(x, y, z) - H\_LUV(x-1, y, z)| + \\ |H\_LUV(x, y, z) - H\_LU(x, y-1, z)| + \\ |H\_LUV(x, y, z) - H\_LU(x, y, z-1)| \end{array} \right) \quad (18),$$

where GL is the number of bins for each component in the H\_LUV color histogram. The algorithm in equation (18) is derived from the 3-D color discreteness algorithm for any color space, defined for a generic color space as follows

$$R = \sum \left( \begin{array}{l} |H(x, y, z) - H(x-1, y, z)| + \\ |H(x, y, z) - H(x, y-1, z)| + \\ |H(x, y, z) - H(x, y, z-1)| \end{array} \right) \quad (19).$$

[0044] The image type decision 418 compares the results of the 3-D color discreteness algorithm (i.e., 414) to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of a 3-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124) by the 3-D color discreteness feature (i.e.,  $R_{LUV}$ ).

[0045] With reference to FIG. 6, a flowchart of an image classification process 500 using a combination of 1-D color discreteness features, 2-D color discreteness features, and 3-D color discreteness features in an embodiment of the invention is shown. Notably, this color discreteness combination classifier includes features of the three classifiers using 1-D color discreteness features, 2-D color discreteness features, and 3-D color discreteness features discussed above. By combining the three color discreteness classifiers, performance may be improved over an individual color discreteness classifier. The color discreteness combination classifier can take advantage of reduced processing overhead by making an early classification, for example, using the 1-D color discreteness features where the classification is more obvious. Conversely, the color discreteness combination classifier can take advantage of increased precision by continuing to analyze the statistics of the input image using 2-D color discreteness features and 3-D color discreteness features.

[0046] Like the individual processes 200, 300, 400, the process 500 begins with an input image 202, the input image is transformed into a color space 204, and the image is smoothed using an averaging filter 206. Again, although CIELUV space is used in the embodiment being described, many other color spaces can also be used.

[0047] The process 500 continues by performing steps 208 through 216 of the 1-D color discreteness classifier 508 shown in FIG. 3. The image type decision 510

compares the results of the 1-D color discreteness algorithms (i.e., 212, 214, or 216) selected for performance to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of the selected 1-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 512) by the 1-D color discreteness feature (i.e.,  $R_L$ ,  $R_U$ , or  $R_V$ ). Alternatively, the 1-D color discreteness classifier may use all three 1-D color discreteness features in any sequence, any combination of two features in any sequence, or any one feature during step 508

[0048] If the result of the 1-D color discreteness classifier is indeterminate 512, the process 500 continues by performing steps 308 through 316 of the 2-D color discreteness classifier 514 shown in FIG. 4. The image type decision 516 compares the results of the 2-D color discreteness algorithms (i.e., 312, 314, or 316) selected for performance to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of a 2-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 518) by the 2-D color discreteness feature (i.e.,  $R_{LU}$ ,  $R_{LV}$ , or  $R_{UV}$ ). Alternatively, the 2-D color discreteness classifier may use all three 2-D color discreteness features in any sequence, any combination of two features in any sequence, or any one feature during step 514

[0049] If the result of the 2-D color discreteness classifier is indeterminate 518, the process 500 continues by performing steps 408 through 414 of the 3-D color discreteness classifier 520 shown in FIG. 5. The image type decision 522 compares the results of the 3-D color discreteness algorithm (i.e., 414) to previously selected thresholds (e.g., low threshold ( $T_L$ ) and high threshold ( $T_H$ )). If the result of a 3-D color discreteness algorithm is above  $T_H$  or below  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124) by the 3-D color discreteness feature (i.e.,  $R_{LUV}$ ).

[0050] Alternatively, the color discreteness combination classifier may perform the 1-D color discreteness classifier, the 2-D color discreteness classifier, and 3-D color discreteness classifier in any sequence and may perform all three color discreteness classifiers or any two color discreteness classifiers before making an image type decision.

[0051] With reference to FIG. 7, a flowchart of an image classification process 600 using edge features in an embodiment of the invention is shown. The process 600 begins with an input image 602. First, edges of color areas in the image are detected 604 using a standard Canny edge detector and an edge map image is created. The parameters identified for the edge detector were determined empirically. Deviations that produce suitable results are also contemplated. Next, the edges in the edge map image are connected 606 (e.g., using a standard 8-connected component algorithm). The average number of pixels per connected edge (P/E) in the edge map image is used as a feature 608. The algorithm for this edge feature is defined as.

$$P/E = \frac{\text{No. of Edge Pixels}}{\text{No. of Connected Edges}} \quad (20).$$

[0052] Typically, synthetic graphic images have fewer connected edges, but each connected edge includes a large number of pixels. On the other hand, pictures have a lot more connected edges, but usually very few pixels in each connected edge. This feature is particularly accurate for high values. In other words, if the value of P/E is high, it is almost certain that the image is synthetic graphic image. However, if the value of P/E is low, the classification of the image is less certain (e.g., indeterminate). This is because the P/E value may be low for synthetic graphic images that have low frequency halftones or certain types of backgrounds. Accordingly, the image type decision 610 compares the result of the P/E feature algorithm to a previously selected high threshold (i.e.,  $T_{HI}$ ). If the result exceeds the high threshold ( $T_{HI}$ ), the classification is synthetic graphic 126. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124).

[0053] The orientation of edges (EO) in the edge map image is also used as a feature 609. As shown in FIG. 7, the connected edges 606 in the edge map image are used to compute an edge orientation histogram 607. For example, the Hough Transform may be used to compute the edge orientation histogram. The Hough Transform is a well-

known method for finding lines. A detailed description of the Hough Transform can be found in "Digital Picture Processing", by Azriel Rosenfeld and Avinash C. Kak, (Academic Press, Inc, 1982) Vol. 2, pp 121-126. The Hough Transform converts an edge map image into a 2-D histogram with one dimension being the line orientation and the other being the line intercept. A Hough Transform entry  $HT(x, y)$  represents the length of a line that has an orientation of  $x$  and an intercept of  $y$ . The edge orientation histogram  $H(x)$  607 can be obtained by manipulating the  $HT(x, y)$  histogram as follows:

$$H(x) = \left( \sum HT(x, y)^2 \right)^{\frac{1}{2}} \quad (21),$$

where the summation is over all  $y$  values.

[0054] The edge orientation (EO) algorithm 609 is performed on the edge orientation histogram  $H(x)$  607 as follows:

$$EO = - \sum H(x) \log H(x) \quad (22)$$

[0055] It has been observed that synthetic graphic images often contain many long straight edges that are oriented in the same direction, typically horizontal and vertical directions, while the edges in natural picture images are typically shorter and oriented more randomly. If many of the edge pixels have a common orientation, the resulting histogram will show one or more spikes. The result of the edge orientation algorithm (i.e., entropy measure) for a spiky histogram is a low number. Accordingly, the image type decision 612 compares the result of the EO feature algorithm 609 to a previously selected low threshold (i.e.,  $T_L$ ). If the result is less than the low threshold ( $T_L$ ), the classification is synthetic graphic 126. Otherwise, the class of the image cannot be determined (i.e., indeterminate 124).

[0056] Furthermore, it is understood that additional edge features may be used for classification of images between natural picture and synthetic graphic classes. Any combination of edge features can also be used by the classifier in any sequence.

[0057] With reference to FIG. 8, a flowchart of an image classification process 700 using a combination of SGLD texture features, color discreteness features, and edge features in an embodiment of the invention is shown. Notably, this image classifier combines all the features of the three types of classifiers discussed above, including 1-D, 2-D, and 3-D features in regard to color discreteness. By combining

SGLD texture, color discreteness, and/or edge features in one classifier, performance may be improved over classifiers using a single feature.

[0058] The process 700 begins with an input image 702. Next, the features are extracted from the input image 704. Feature extraction includes compiling SGLD texture features 706 (e.g., variance (V), bias (B), skewness (S), fitness (F)), color discreteness features 708, including 1-D color discreteness features, (e.g., R\_L, R\_U, R\_V), 2-D color discreteness features, (e.g., R\_LU, R\_LV, R\_UV), and 3-D color discreteness features, (e.g., R\_LUV), and edge features 710 (e.g., P/E, EO).

[0059] The SGLD texture features are extracted by performing steps 104-110 of the process 100 depicted in FIG. 1. Similarly, the 1-D color discreteness features are extracted by performing steps 204-211 of the process 200 depicted in FIG. 3. Likewise, the 2-D color discreteness features are extracted by performing steps 204-311 of the process 300 depicted in FIG. 4. Similarly, the 3-D color discreteness features are extracted by performing steps 204-411 of the process 400 depicted in FIG. 5. Likewise, the edge features are extracted by performing steps 604-607 of process 600 depicted in FIG. 7.

[0060] Next, the feature algorithms are performed 716. The SGLD texture feature algorithms are performed by accomplishing steps 112-118 of the process 100 depicted in FIG. 1. Similarly, the 1-D color discreteness feature algorithms are performed by accomplishing steps 212-216 of the process 200 depicted in FIG. 3. Likewise, the 2-D color discreteness feature algorithms are performed by accomplishing steps 312-316 of the process 300 depicted in FIG. 4. Similarly, the 3-D color discreteness feature algorithm is performed by accomplishing step 414 of the process 400 depicted in FIG. 5. Likewise, the edge feature algorithms are performed by accomplishing steps 608 and 609 of the process 600 depicted in FIG. 7.

[0061] Finally, the image type decision 718 compares the results of the feature algorithms (e.g., V, B, S, F, R\_L, R\_U, R\_V, R\_LU, R\_LV, R\_UV, R\_LUV, P/E, EO) to previously selected thresholds (e.g., low thresholds ( $T_L$ ) and/or high thresholds ( $T_H$ )) associated with each feature. If the result of any feature algorithm is above the associated  $T_H$  or below the associated  $T_L$ , the image is classified as either a synthetic graphic 126 or natural picture 122 according to predetermined rules. Otherwise,

the class of the image cannot be determined (i.e., indeterminate 124) by the combination classifier. The combination classifier may also incorporate logic to resolve conflicting results, particularly where one individual classifier is indeterminate 124 and another individual classifier classifies the image as either a synthetic graphic 126 or natural picture 122.

[0062] The embodiment described and shown in FIG. 8 uses all features available to the combination classifier and performs feature extraction 704 in a parallel fashion, performs feature algorithms 716 in a parallel fashion, and then makes a combination image type decision 718. Alternatively, the combination classifier may use all features available to it in any sequence. In other words, feature extraction 704, feature algorithms 716, and image type decisions 718 can be performed for individual features, in a serial fashion, similar to that described above for the color discreteness combination classifier shown in FIG. 6. Further alternatives, using a serial-parallel combination in any order with any combination of any two or more features in parallel legs of such a combination classifier are also contemplated. Many other alternative combination classifiers are also available by using any combination of two or more of the features available to the combination classifier in parallel fashion, serial fashion, or serial-parallel fashion.

[0063] With reference to FIG. 9, a block diagram of an image segmentation system 800 using a “binary” image classification process (i.e., classification of images between natural picture or synthetic graphic classes) is shown. The picture/graphic classifiers (i.e., 100, 200, 300, 400, 500, 600, 700) of FIGS 1-8 are “binary” classifiers and could be implemented in such a system 800. As described above for FIGS. 1-8, an input image 802 is provided to a feature extractor 804. The feature extractor 804 extracts pertinent characteristics (i.e., features) based on the parameters required by algorithms of the binary classifier 806. The binary classifier 806 exercises algorithms designed to classify the input image between a natural picture or a synthetic graphic image (e.g., [0, 1] where 0 indicates natural picture and 1 indicates synthetic graphic). This binary classification result is provided to a switch 808. The switch 808 receives the input image 802 and switches it between natural picture processing 810 and synthetic graphic processing 812, depending on the binary classification result.

[0064] Natural picture processing 810 processes the image in a manner tailored to maximize the quality of natural picture images (e.g., gamut mapping) Similarly, synthetic graphic processing 812 is tailored to maximizes the quality of synthetic graphic images (e.g., filtering). If the input image is classified as a picture, the input image 802 is switched to natural picture processing 810 and a picture output 814 is produced. Alternatively, if the image is classified as a synthetic graphic, the input image 802 is switched to synthetic graphic processing 812 and a synthetic graphic output 816 is produced. In the event that the binary classifier 806 cannot determine the class of the input image, one of the processes (e.g., natural picture processing 810) may be selected by default.

[0065] The invention has been described above with reference to various embodiments. Obviously, modifications and alterations will occur to others upon reading and understanding the preceding detailed description. It is intended that the invention be construed as including all such modifications and alterations insofar as they come within the scope of the appended claims or the equivalents thereof